日　本　国　特　許　庁
JAPAN　PATENT　OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:　２００２年　９月１１日

出願番号
Application Number:　特願２００２－２６４９７７

[ST.10/C]:　[ＪＰ２００２－２６４９７７]

出願人
Applicant(s):　株式会社東芝

２００３年　２月１４日

特許庁長官
Commissioner,
Japan Patent Office　太田信一郎

出証番号　出証特２００３－３００７４６３

【書類名】　特許願

【整理番号】　13B0280691

【あて先】　特許庁長官殿

【国際特許分類】　G09C　1/00

【発明の名称】　暗号演算回路

【請求項の数】　5

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

【氏名】　藤崎　浩一

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

【氏名】　新保　淳

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

【氏名】　本山　雅彦

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

【氏名】　池田　華恵

【発明者】

【住所又は居所】　神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

【氏名】　友枝　裕樹

【特許出願人】

【識別番号】　000003078

【氏名又は名称】　株式会社　東芝

出証特２００３－３００７４６３

【代理人】

【識別番号】　100083161

【弁理士】

【氏名又は名称】　外川　英明

【電話番号】　(03)3457-2512

【手数料の表示】

【予納台帳番号】　010261

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　暗号演算回路

【特許請求の範囲】

【請求項１】　所定ビット数の第１データブロックと鍵情報とを所定の非線形関数Ｆへ作用させた結果得られる結果値と該所定ビット数の第２データブロックとを論理演算して結果値を得て、前記第２データブロックを第１データブロックと置き換えるとともに、前記第１データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、

前記非線形関数Ｆを作用させるための前記鍵情報を前記暗号化処理とは無関係の情報として供給する手段を備え、前記暗号化データの出力の後に、前記供給する手段によって開始するようにしたことを特徴とする暗号演算回路。

【請求項２】　所定ビット数の第１データブロックと鍵情報とを所定の非線形関数Ｆへ作用させた結果得られる結果値と該所定ビット数のデータブロックとを論理演算して結果値を得て、前記第２データブロックを第１データブロックと置き換えるとともに、前記第１データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、

前記第1データブロックを保持する保持手段と、

この保持手段へ前記暗号化処理とは無関係の情報を供給する手段を備え、前記暗号化データの出力の後には、前記供給する手段によって供給を開始するようにしたことを特徴とする暗号演算回路。

【請求項３】　所定ビット数の第１データブロックと鍵情報とを所定の非線形関数Ｆへ作用させた結果得られる結果値と該所定ビット数の第２データブロックとを論理演算して結果値を得て、前記第２データブロックを第１データブロックと置き換えるとともに、前記第１データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、

前記非線形関数Ｆを作用させるための前記鍵情報を前記暗号化処理とは無関

係の情報として供給する第１供給手段と、

前記第１データブロックを保持する保持手段と、

この保持手段へ前記暗号化処理とは無関係の情報を供給する第２供給手段とを備え、

前記暗号化データの出力の後には、前記第１供給手段および前記第２供給手段によってそれぞれ供給を開始するようにしたことを特徴とする暗号演算回路。

【請求項４】　非線形関数Ｆを処理する手段を備えるＦｅｉｓｔｅｌ型暗号アルゴリズムを実装し、暗号化データを出力する暗号演算回路であって、

前記非線形関数Ｆを処理する手段へ暗号演算結果とは無関係のデータを供給する手段を備え、前記暗号化データの出力の後に、前記供給する手段によってデータ供給を開始するようにしたことを特徴とする暗号演算回路。

【請求項５】　Ｆｅｉｓｔｅｌ型暗号アルゴリズムを実装し、暗号化データを出力する暗号演算回路であって、

暗号演算回路内の内部データを保持する保持手段と、

この保持手段へ前記暗号化処理とは無関係の情報を供給する手段とを備え、

前記暗号化データの出力の後には、前記供給する手段によって供給を開始するようにしたことを特徴とする暗号演算回路。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、Ｆｅｉｓｔｅｌ型アルゴリズムを実装した暗号演算回路に関し、特に、電力解析攻撃に対する耐性を持った暗号化演算回路に関する。

【０００２】

【従来の技術】

暗号演算回路に対して演算処理中の消費電力に関するグラフを作成し、その消費電力グラフを用いて、暗号演算回路に納められている鍵情報を取り出すという電力解析攻撃(ＤＰＡ: Differential Power Analysis) 法がある。

【０００３】

ＤＰＡ法は、まず、複数のデータをそれぞれ暗号演算回路に入力し、それぞれ

の消費電力を測定する。次に、暗号演算回路内にある鍵情報の推定を行う。このとき、推定した鍵情報と消費電力グラフに相関がある場合、計算されたその相関値は、推定した鍵情報が正しい時には大きな値となり、推定した鍵情報が間違っている時には小さな値となる。ＤＰＡ法は、この原理を用いて、消費電力を測定することにより暗号演算回路内部の鍵情報を取り出すという攻撃方法である。

【０００４】

ＤＰＡ法は、破壊行為を伴わない攻撃であるため、外見を見ただけでは攻撃によって鍵情報が取り出されたかどうか判断できず、不正利用による被害が拡大する恐れがある。このため、暗号演算回路において、ＤＰＡ法への対策が必須となってきている。

【０００５】

ＤＰＡ法に対する対策の一つとして、演算中のデータをマスクして演算する発明があった（例えば、特許文献１参照）。この発明は、鍵情報を用いた演算中の消費電力の変動が鍵情報と関係しないように、鍵情報をマスクして演算を行なうというものである。この発明は、鍵情報をマスクして演算することで、鍵情報と消費電力との相関関係を失わせるようにして、ＤＰＡ法対策を行ったものである。

【０００６】

【特許文献１】特開２０００－６６５８５公報

【０００７】

【発明が解決しようとする課題】

ところで、共通鍵ブロック暗号方式では、Feistel氏の開発したFeistel型暗号アルゴリズムを用いたものが多く採用されている。

【０００８】

従来の技術欄で説明した、特開２０００－６６５８５公報の発明は、このFeistel型暗号アルゴリズムを実装した暗号演算回路において、演算終了時点のＤＰＡ法の対策として有効であるが、演算終了の後の回路動作に対して行われるＤＰＡ法に対する対策では無かった。

【０００９】

そこで、本発明は、このような問題点に鑑みなされたものであり、Feistel型暗号アルゴリズムを実装した暗号演算回路の演算終了の後に行われるＤＰＡ法の対策を実現した暗号演算回路を提供することを目的とする。

【００１０】

【課題を解決するための手段】

上記目的を達成するために、本発明は、所定ビット数の第１データブロックと鍵情報とを所定の非線形関数Ｆへ作用させた結果得られる結果値と該所定ビット数の第２データブロックとを論理演算して結果値を得て、前記第２データブロックを第１データブロックと置き換えるとともに、前記第１データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、前記非線形関数Ｆを作用させるための前記鍵情報を前記暗号化処理とは無関係の情報として供給する手段を備え、前記暗号化データの出力の後に、前記供給する手段によって始するようにした。

【００１１】

また、本発明は、所定ビット数の第１データブロックと鍵情報とを所定の非線形関数Ｆへ作用させた結果得られる結果値と該所定ビット数のデータブロックとを論理演算して結果値を得て、前記第２データブロックを第１データブロックと置き換えるとともに、前記第１データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、前記第1データブロックを保持する保持手段と、この保持手段へ前記暗号化処理とは無関係の情報を供給する手段を備え、前記暗号化データの出力の後には、前記供給する手段によって供給を開始するようにした。

【００１２】

本発明は、例えば、Feistel型暗号アルゴリズムを実装した暗号演算回路の信号線の遷移や信号線の値に着目し、それぞれにおいてＤＰＡ法により攻撃されうるタイミングと攻撃ポイントを明確にして、上記のように対策するようにしたから、ＤＰＡ法による攻撃に対して暗号演算回路の内部にある鍵情報の漏洩を防ぐ

ことができる。

【００１３】

【発明の実施の形態】

以下、本発明の実施の形態について、図面を用いて詳細に説明する。

【００１４】

まず、Feistel型暗号アルゴリズムについて、その処理の流れを図１のフローチャートを用いて簡単に説明する。

【００１５】

まず、暗号化する入力ブロックデータは、初期転値(ＩＰ:Initial Permutation)と呼ばれる、入力ブロックデータのビット間のデータを並び換える処理が施される（ステップＳ１）。次に、この初期転値後のブロックデータをｎ／２ビットのブロックに分割する（ステップＳ２）。この分割された２つのブロックを初期値とし、以下の演算を繰り返す（ステップＳ３、Ｓ４）。

$L_i = R_{i-1}$

$R_i = L_{i-1} \hat{} F(key_i, R_{i-1})$

ここで、演算回数をｉ（１≦ｉ≦ｋ)、ｉ回目の演算時の一方の分割されたブロック（右側）をR_i、他方のブロック（左側）をL_i、ｉ回目の鍵情報をkey_i、非線形演算を行なう関数をＦとする。また、「＾」は、排他的論理和を示す。

【００１６】

この繰り返し演算Ｓ３をｋ回繰り返した後（Ｓ４のＹ）に、各演算結果R_k、L_kを合成する（ステップＳ５）。なお、この合成は、L_kを上位ｎ／２ビットのブロックに、R_kを下位ｎ／２ビットのブロックとしたｎビットのブロックデータとすることを意味する。そして、合成されたブロックデータに対し、逆初期転値（IP^{-1}）を行ない（ステップＳ６）、演算結果を得て演算を終了する。

【００１７】

次に、このFeistel 型暗号アルゴリズムを実装した暗号演算回路のデータパス部の構成を図２に示す。

【００１８】

ＩＰ部２１は、予め定められた規則に基づいて、初期転値の演算を行なう回路

モジュールである。関数演算部２４は、所定の非線形関数である関数Ｆの演算を行なう回路モジュールである。ＩＰ$^{-1}$部２５は、ＩＰ部２１で定められた規則に相反する規則に基づいて、逆初期転値の演算を行なう回路モジュールである。Ｌレジスタ２３、Ｒレジスタ２２は、それぞれｎ／２ビット幅を持ったレジスタであり、３入力１出力のセレクタ２６、２７は、図示されていない制御回路部からの信号に応じて、Ｌレジスタ２３、Ｒレジスタ２２へ入力するデータを選択する。なお、セレクタ２６への３入力とは、ＩＰ部２１からの上位ｎ／２ビットの値と、Ｒレジスタ２２の出力値と、Ｌレジスタ２３の出力値と関数演算部の出力値とを排他的論理和した値とであり、セレクタ２７への３入力とは、ＩＰ部２１からの下位ｎ／２ビットの値と、Ｒレジスタ２２の出力値と、Ｌレジスタ２３の出力値と関数演算部の出力値とを排他的論理和した値とである。

【００１９】

さて、以上説明した構成において処理された暗号演算結果Ｃ＝（C_l、C_r）は、次式のように表すことができる。

$C = IP^{-1}\{(L_k \hat{} F(key_k、R_k))、R_k\}$

ここで、ＩＰ$^{-1}$部２５は、ビット並びを入れ換えているだけであるために、演算結果ＣからR_kの値は容易に特定される。R_kの値が特定されると、演算終了後の関数演算部２４の出力値は、鍵情報key_kの一変数の値のみによって決まる。以上のことから、関数演算部２４の出力結果を攻撃ポイントとして、鍵情報key_kを推定することにより、ＤＰＡ法が行われてしまう。仮に、関数演算部２４の消費電力の変動と鍵情報の演算との間に相関があった場合には、攻撃者に鍵情報を特定されてしまう。すなわち、暗号演算の終了時点で、ＤＰＡ法を用いた攻撃を行なうことで、暗号演算回路内の鍵情報が取り出されてしまう。このＤＰＡ法を用いた攻撃法の対処方法としては、先に示した特開２０００－６６５８５号公報による発明によって解決できる。

【００２０】

一方、Feistel型暗号アルゴリズムを回路実装した暗号演算回路においては、上記の暗号演算の終了時点での攻撃の他に、暗号演算が終了後（終了時点の次のタイミング以降）においても、ＤＰＡ法を用いた攻撃を行うことで、暗号演算回

路内の鍵情報が取り出されてしまう可能性があることが新たに分かった。このことを説明する前に、暗号演算回路の演算結果を出力する際のデータパス部におけるデータの制御方法には２通り有って、各制御方法によって、暗号演算の終了後のＤＰＡ法による攻撃ポイントが異なっているため、まず、２通りのデータの制御方法について説明する。

【００２１】

１つめの制御方法は、演算結果を出力するタイミングにおいて、図３にあるようにセレクタ２７がデータパス２８を選択し、かつセレクタ２６がデータパス２９を選択することで、左右のデータを入れ換えた形で演算を終わらせる、クロス型回路構成である。一方、２つ目の制御方法は、演算結果を出力するタイミングにおいて、図１０にあるようにセレクタ２７がデータパス２９を選択し、セレクタ２６がデータパス２８を選択することで、演算の最後において左右のデータを入れ換えないような形で演算を終わらせる、ストレート型回路構成である。

【００２２】

Feistel型暗号アルゴリズムを実装した暗号演算回路において、そのアルゴリズムで定められている繰り返し演算を行なう場合、セレクタ２６、セレクタ２７は左右のデータが入れ替わるように、図２において図示されていない制御回路により制御される。この暗号演算回路において、演算結果を出力する場合に、特に必要がない場合は繰り返し演算を行なう場合と同様に、左右のデータが入れ替わったクロス型回路構成で演算を終了させる。このようなクロス型回路構成とする理由として、演算結果を出力するタイミング演算結果を出力するタイミングでデータの流れを変えるように制御するよりも、演算結果を出力する場合であっても、繰り返し演算を行なっているときと同様に左右のデータが入れ替わるような制御を行なった方が、制御が簡単であり制御回路の実装も小さくできるからである。

【００２３】

一方、ＤＥＳを３回行なうTriple DES(3DES)を実装した回路の場合には、１回目のＤＥＳの演算結果と２回目のＤＥＳの演算開始間、および２回目のＤＥＳの演算結果と３回目のＤＥＳの演算開始間において、左右のデータを入れ換えない

ような制御を行なう必要がある。そのため、３回目のＤＥＳが終了後も左右のデータを入れ換えずに出力するストレート型回路構成とすることが多い。このような回路構成となるのは、暗号演算回路として回路実装する場合において、その回路規模を小さくするために、演算中にできる限り場合分けを減らすような制御を行なうようにするためである。１回目、２回目が終わったときと同様に、３回目が終わったときにもストレート型回路構成となるようにセレクタ２６、セレクタ２７を制御したほうが、場合分けの数が少なくなり制御回路を小さく構成することができる。

【００２４】

以上説明したように、Feistel 型暗号アルゴリズムを実装した場合には、クロス型回路構成とストレート型回路構成との何れかで構成される。

【００２５】

次に、これら回路構成の違いによる演算終了後におけるＤＰＡ法による攻撃ポイントについて説明する。

【００２６】

まず、クロス型回路構成の場合について、ＤＰＡ法により攻撃のポイントとなるタイミングと、その原因について説明する。

【００２７】

演算結果を出力するクロック$Clock_{fin}$の立ち上がりのタイミングにおいて、Ｌレジスタ２３の値をL_{fin}、Ｒレジスタ２２の値をR_{fin}とする。このとき、この暗号演算回路の演算結果をＣとすると、ＣはC_lとC_rを用いて$C=IP^{-1}(L_{fin} \hat{} F(key_{fin}, R_{fin}), R_{fin})$と表すことができる。ここで、$IP^{-1}$は、ビットの並びを変換する処理だけであるため、演算結果ＣからR_{fin}と$L_{fin} \hat{} F(key_{fin}, R_{fin})$の値は簡単に解ってしまう。

【００２８】

次に、クロック$Clock_{fin+1}$において、Ｒレジスタ２２に値が書き込み可能である場合、Ｒレジスタ２２の値は$L_{fin} \hat{} F(key_{fin}, R_{fin})=R_{fin+1}$となる。このとき、鍵情報が$Clock_{fin}$と同じ$key_{fin}$であるので、関数演算部２４の出力は$F(key_{fin}, R_{fin+1})$となる。$R_{fin+1}$は、暗号演算回路の出力結果から簡単に解っ

てしまう値であり、そのため、関数演算部２４の関数Ｆは、key_{fin}の一変数の関数と見なすことができる。

【００２９】

以上のように、関数演算部２４の出力の遷移を攻撃ポイントとして、関数演算部２４の出力を推定することにより鍵情報を特定されてしまうといった問題点があった。

【００３０】

また、上記の他に、Ｌレジスタ２３の値の遷移を攻撃ポイントとされる別の問題点がある。これについて説明する。

【００３１】

$Clock_{fin}$、$Clock_{fin+1}$におけるＬレジスタ２３の値をL_{fin}、L_{fin+1}とする。Ｌレジスタ２３の値の遷移は次式で表すことができる。

$L_{fin} \hat{} L_{fin+1}=R_{fin-1} \hat{} R_{fin}$

なお、上記式の右辺は、クロス型回路構成においてＬレジスタ２３の値は$L_i=R_{i-1}$($1 \leqq i \leqq k$)という関係があることから、書き変えたものである。ここで、R_{fin}は外部から観測可能な値であることから、R_{fin-1}に対して推定することができる。Ｌレジスタ２３の遷移は、暗号演算回路の消費電力の変動と関係があるので、ＤＰＡ法によりR_{fin-1}の値が特定される可能性がある。つまり、上記で述べたように演算結果Ｃから、$L_{fin} \hat{} F(key_{fin}、R_{fin})(=R_{fin+1})$および$R_{fin}$の値は観測可能である。ここで、$L_{fin}=R_{fin-1}$であることから、$R_{fin-1}$が特定されると、$R_{fin+1}$、$L_{fin}$、$R_{fin}$が知ることのできる値となるため、$key_{fin}$を特定されてしまうという問題点があった。

【００３２】

以上のように、関数演算部２４の出力値の遷移、または、Ｌレジスタ２３の値の遷移において、ＤＰＡ法を用いて攻撃される可能性がある。

【００３３】

クロス型回路構成の場合のこれらＤＰＡ法による攻撃に対する対策は、それぞれ異なる。

【００３４】

まず、このクロス型回路構成の$Clock_{fin+1}$のタイミングにおける関数演算部２４の出力値に関するＤＰＡ対策として、本実施の形態においては、$Clock_{fin+1}$のときに関数演算部２４に入力される値は、鍵情報と関係のない値を入力するような回路構成とする。このような回路の具体例としては、図４に示すように関数演算部２４へ鍵情報を入力する個所へ新たにセレクタを設け、$Clock_{fin}$までは鍵情報を入力し、$Clock_{fin+1}$以降は乱数を入力するような回路構成や、図５にあるように関数演算部２４の入力として、鍵情報と乱数を排他的論理和（ＸＯＲ）演算、論理積（ＡＮＤ）演算、または論理和（ＯＲ）演算した値を入力する回路構成とし、$Clock_{fin+1}$以降のクロックの立ち上がり時に、常に異なる乱数を与えて得た演算結果を入力するという回路構成で解決した。

【００３５】

また、関数演算部２４の入力として、外部から観測可能な値を入れないようにする回路構成も対策となりうる。このような対策の具体例としては、図６に示すように、関数演算部２４の入力の前にセレクタを設け、$Clock_{fin+1}$以降は乱数を入力するような回路構成を用いる例がある。また、図６のセレクタを用いて乱数を入れる代わりに、図７のように演算回路を追加し、Ｒレジスタ２２の値と乱数との演算結果が、関数演算部２４に入力されるような回路構成としてもよい。このように$Clock_{fin+1}$以降、関数演算部２４の入力として、暗号演算回路で利用した鍵情報や演算結果が、そのまま入力されないような回路構成とすることで、関数演算部２４の出力におけるＤＰＡ対策とすることができる。

【００３６】

このほかに、$Clock_{fin+1}$以降、Ｒレジスタ２２の値を、演算結果と無関係な値とすることで、ＤＰＡ法対策ができる。例えば、図８にあるようにＲレジスタ２２の入力の前にセレクタを入れ、$Clock_{fin+1}$以降、このセレクタから乱数がＲレジスタ２２に書き込まれるような回路構成や、図６のようにデータパス２９の結果と乱数を演算するような回路構成とすればよい。

【００３７】

以上のような対策を施した暗号演算回路とすることにより、関数演算部２４の出力の遷移を攻撃のポイントとできなくなった。

【００３８】

次に、Ｌレジスタ２３の値の遷移を攻撃ポイントとした場合には、$Clock_{fin+1}$のタイミングでＬレジスタ２３にR_{fin+1}を書き込まないような回路構成としなければならない。この時の対策として、回路構成の具体例は、図１１のようにＬレジスタ２３へのデータパス３０上にＡＮＤ／ＯＲ／ＸＯＲなどの演算回路を入れ、$Clock_{fin+1}$以降でデータパス３０上の値と乱数の演算結果をＬレジスタ２３に書き込むという方法や、図１２にあるようにデータパス３０にセレクタを入れることで$Clock_{fin+1}$以降、Ｌレジスタ２３に乱数が書き込まれるような回路構成で実現できる。

【００３９】

以上のような対策を施した暗号演算回路とすることにより、Ｌレジスタ２３の値の遷移を攻撃のポイントとできなくなった。

【００４０】

次に、図１０のストレート型回路構成の場合についてのＤＰＡ法による攻撃のポイントとなるタイミングとその原因について説明する。

【００４１】

ストレート型回路構成においてもクロス型回路構成のときと同様に、演算結果を出力するときのクロック$Clock_{fin}$の立ち上がりのタイミングにおいて、Ｌレジスタ２３の値をL_{fin}、Ｒレジスタ２２の値をR_{fin}とする。$Clock_{fin}$の次のクロック$Clock_{fin+1}$の立ち上がりのタイミングにおいて、Ｌレジスタ２３、Ｒレジスタ２２の値はそれぞれL_{fin+1}、R_{fin+1}とすると、次式で表すことができる。

$L_{fin+1}=L_{fin} \hat{} F(key_{fin}, R_{fin})$

$R_{fin+1}=R_{fin}$

この$Clock_{fin+1}$のタイミングにおいて、Ｌレジスタ２３の値の遷移に着目すると、Ｌレジスタ２３の値の遷移は$L_{fin} \hat{} L_{fin+1}=F(key_{fin}, R_{fin})$と表すことができる。Ｌレジスタ２３の値の遷移$F(key_{fin}, R_{fin})$は、鍵情報に関係する情報を含んでいるうえに、$R_{fin}$は外部から観測可能な値である。従って、Ｌレジスタ２３の値の遷移に対し、ＤＰＡ法を用いて攻撃が行われると、暗号演算回路の鍵情報key_{fin}が特定されてしまう。

【００４２】

そのため、本実施の形態においては、Ｌレジスタ２３の値が、$Clock_{fin+1}$において暗号演算回路が持つ鍵情報と相関を持つことのないような回路構成とする。この回路構成の具体例としては、クロス型回路構成で説明した方法と同様の図１２にあるようにＬレジスタ２３へのデータパス２８上にＡＮＤ、ＯＲ、ＸＯＲなどの論理演算器を配置し、データパス２８の値と乱数の演算結果をＬレジスタ２３に書き込むという方法や、図１１にあるように、$Clock_{fin+1}$のタイミングにて、データパス２８の値と乱数とをセレクタで切り替えるという方法で実現可能である。

【００４３】

また、関数演算部２４に$Clock_{fin+1}$以降、常にkey_{fin}とR_{fin}とが入力されるような回路構成の場合には、上記のクロス型回路構成の時と同様に関数演算部２４の出力値の遷移によって鍵情報が特定される可能性がある。そのため、前記クロス型回路構成の関数演算部２４のＤＰＡ法の対策と同様、図４～図９の何れかを用いることによって、対策を実現できる。

【００４４】

また、クロス型回路構成、ストレート回路構成の何れの場合においても、$Clock_{fin+1}$においてＲレジスタ２２、Ｌレジスタ２３へのデータの書き込みが行なわれず、且つ、$Clock_{fin;1}$以降、常に鍵情報key_{fin}が関数演算部２４に入力されるような回路構成が取られている場合には、関数演算部２４の出力値Ｆ(key_{fin}、R_{fin})となるので、上記のクロス型回路構成の時と同様、関数演算部２４の出力値の遷移がＤＰＡ法による攻撃の対象となってしまう。従って、上記のクロス型回路構成の関数演算部２４のＤＰＡ法の対策と同様、図４～図９の何れかを用いることによって、対策を実現できる。

【００４５】

以上説明してきたように、本実施の形態では、暗号演算回路の演算終了後以降において、鍵情報と消費電力の相関関係を失わせるような回路構成としたから、暗号演算回路内部にある鍵情報の漏洩を防ぐことができ、ＤＰＡ法による攻撃に対する耐性を持った暗号演算回路を提供できるようになった。

【００４６】

【発明の効果】

以上説明したように、本発明によれば、暗号演算回路内部にある鍵情報の漏洩を防ぐことができ、ＤＰＡ法による攻撃に対する耐性を持った暗号演算回路を提供できるようになった。

【図面の簡単な説明】

【図１】Feistel型アルゴリズムを示すフローチャート。

【図２】Feistel型アルゴリズムを採用した暗号演算回路のデータパス部に関する回路構成を示す図。

【図３】クロス型回路構成のデータパス部を示す図。

【図４】本実施の形態における、関数演算部２４のＤＰＡ法対策を施した回路構成の第１例を示す図。

【図５】本実施の形態における、関数演算部２４のＤＰＡ法対策を施した回路構成の第２例を示す図。

【図６】本実施の形態における、関数演算部２４のＤＰＡ法対策を施した回路構成の第３例を示す図。

【図７】本実施の形態における、関数演算部２４のＤＰＡ法対策を施した回路構成の第４例を示す図。

【図８】本実施の形態における、関数演算部２４のＤＰＡ法対策を施した回路構成の第５例を示す図。

【図９】本実施の形態における、関数演算部２４のＤＰＡ法対策を施した回路構成の第６例を示す図。

【図１０】ストレート型回路構成のデータパス部を示す図。

【図１１】本実施の形態における、Ｌレジスタ２３のＤＰＡ法対策を施した回路構成の第１例を示す図。

【図１２】本実施の形態における、Ｌレジスタ２３のＤＰＡ法対策を施した回路構成の第２例を示す図。

【符号の説明】

２１･･･ＩＰ部

２２･･･Ｒレジスタ

２３･･･Ｌレジスタ

２４･･･関数演算部

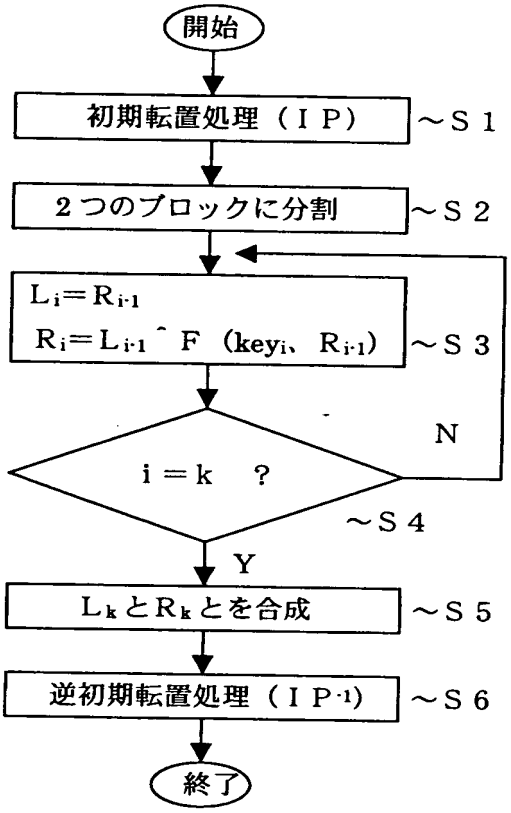
２５･･･$ＩＰ^{-1}$部

２６、２７･･･セレクタ

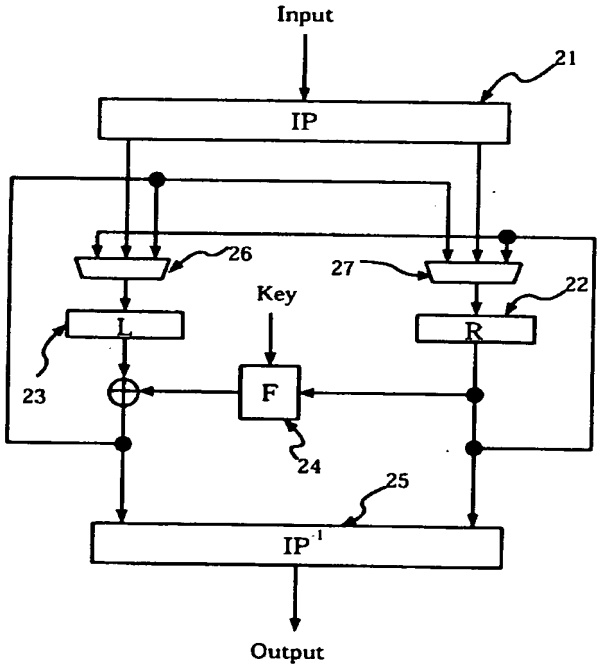
２８、２９、３０･･･データパス

【書類名】　　図面

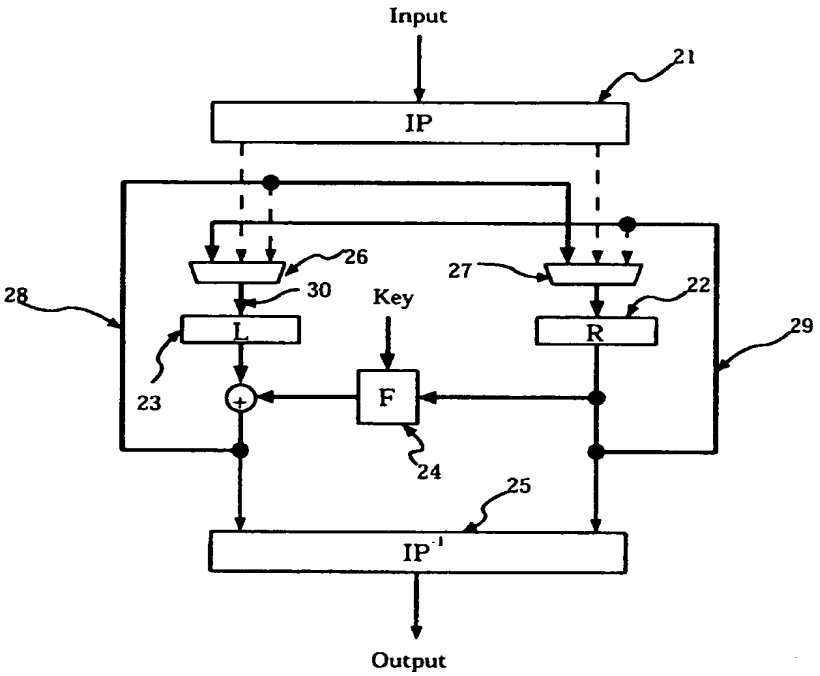
【図１】


開始
初期転置処理（ＩＰ）
～Ｓ１
２つのブロックに分割
～Ｓ２
$L_i = R_{i-1}$
$R_i = L_{i-1} \hat{} F\ (key_i,\ R_{i-1})$
～Ｓ３
N
i＝k　？
～Ｓ４
Y
L_kとR_kとを合成
～Ｓ５
逆初期転置処理（IP^{-1}）
～Ｓ６
終了

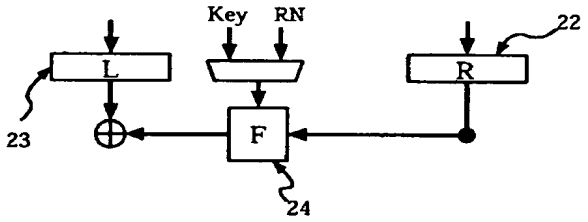
【図２】


Input
21
IP
26
27
Key
22
L
R
23
F
24
25
IP^{-1}
Output

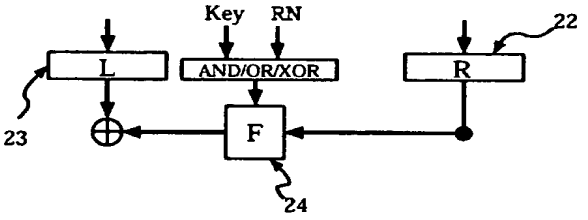
【図３】


Input
21
IP
26
27
30
Key
22
28
L
R
29
23
F
24
25
IP-1
Output

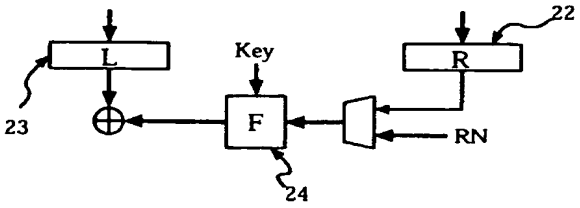
【図４】


Key
RN
22
L
R
23
F
24

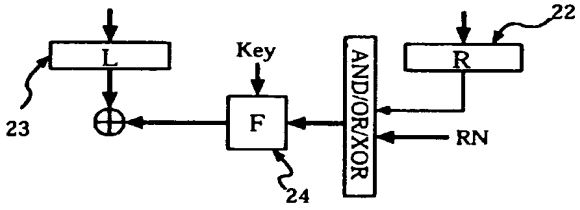
【図５】


Key
RN
22
L
AND/OR/XOR
R
23
F
24

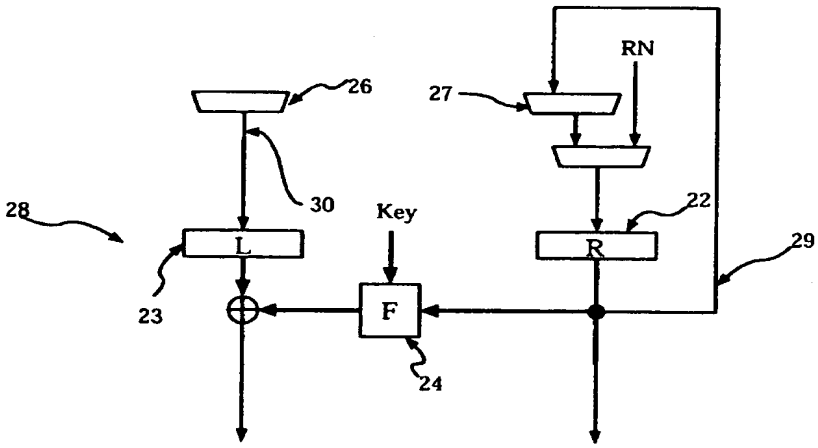
【図６】


L
Key
R
22
23
F
RN
24

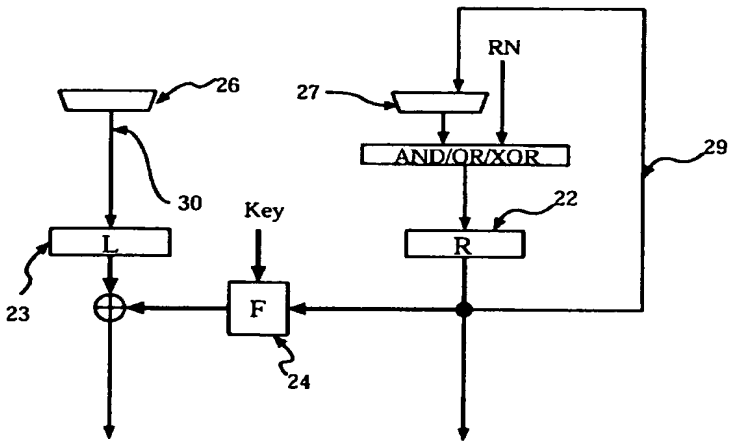
【図７】


L
Key
AND/OR/XOR
R
22
23
F
RN
24

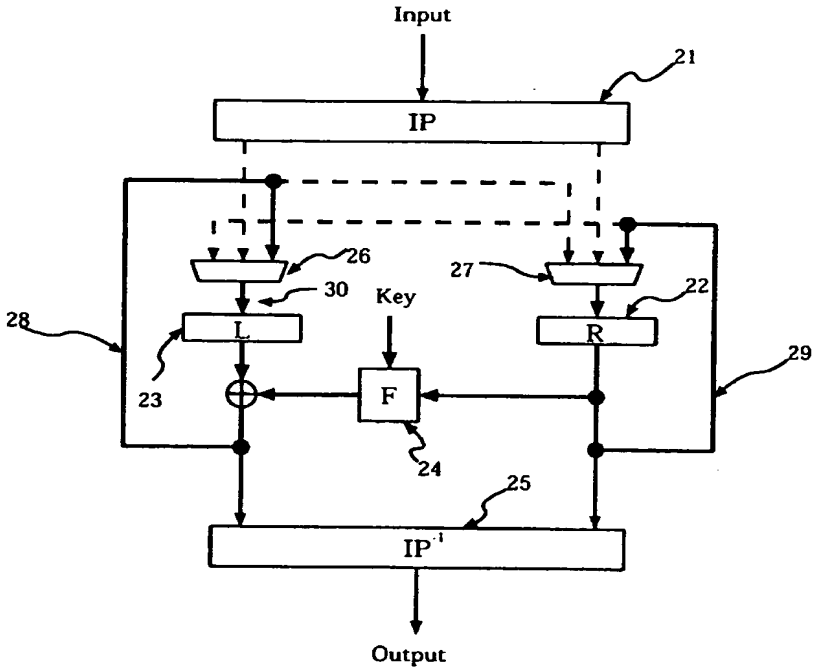
【図８】


RN
26
27
30
28
Key
22
L
R
29
23
F
24

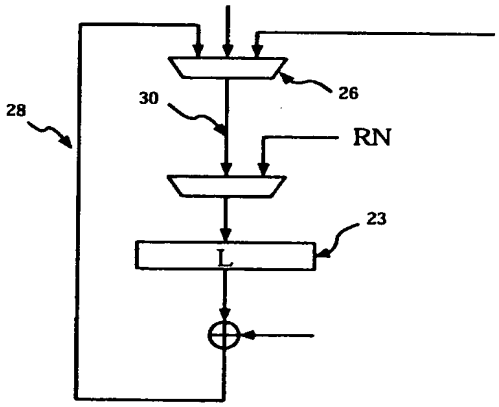
【図９】


RN
26
27
AND/OR/XOR
29
30
Key
22
L
R
23
F
24

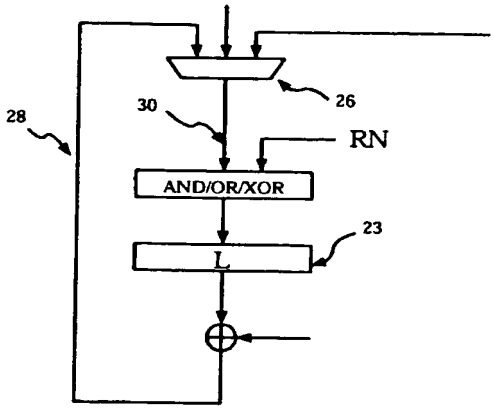
【図１０】


Input
21
IP
26
27
30
Key
22
28
L
R
29
23
F
24
25
IP-1
Output

【図１１】


26
30
28
RN
23
L

【図１２】



【書類名】　要約書

【要約】

【課題】　Ｆｅｉｓｔｅｌ型暗号アルゴリズムを実装した暗号演算回路の演算終了の後に行われるＤＰＡ法の対策を実現した暗号演算回路の提供。

【解決手段】　Ｆｅｉｓｔｅｌ型暗号アルゴリズムを実装した暗号演算回路は、非線形関数Ｆを演算する関数演算部２４と、暗号化データの出力の後に、関数演算部２４へ暗号演算結果とは無関係の乱数データを供給する手段とを備える。

【選択図】　図４

認定・付加情報

特許出願の番号	特願２００２－２６４９７７
受付番号	５０２０１３５７０８６
書類名	特許願
担当官	第七担当上席　００９６
作成日	平成１４年　９月１２日

＜認定情報・付加情報＞

【提出日】　平成14年 9月11日

次頁無

出　願　人　履　歴　情　報

識別番号　　〔０００００３０７８〕

１．変更年月日　　２００１年　７月　２日
［変更理由］　住所変更
住　所　　東京都港区芝浦一丁目１番１号
氏　名　　株式会社東芝